



Universidade Federal de Uberlândia  
Faculdade de Computação  
13º Trabalho de Programação para Internet – Gestão da Informação  
Prof. Daniel A. Furtado – Trabalho Individual  
Desenvolvimento Web com Banco de Dados e Ajax – Parte 2

## Instruções Gerais

---

- Esta atividade deve ser realizada individualmente;
- Utilize apenas as tecnologias HTML5, CSS, JavaScript, Bootstrap 5, PHP, MySQL e XMLHttpRequest e a API Fetch;
- Sintaxe da XHTML como <img/> ou <br/> não é permitida (anulará o trabalho);
- O website deve ser hospedado e disponibilizado online, conforme orientações disponíveis no final deste documento;
- Ao construir o website, utilize dados fictícios (**jamais utilize** dados pessoais como seu nome, CPF, endereço, e-mail etc.);
- Esteja atento às **observações sobre plágio** apresentadas no final deste documento;
- Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet ou de trabalhos de semestres anteriores serão anulados;
- As páginas web não devem conter qualquer conteúdo de caráter imoral, desrespeitoso, pornográfico, discurso de ódio, desacato etc.;
- O website deve ser validado utilizando as ferramentas disponíveis nos endereços **validator.w3.org** e **jigsaw.w3.org/css-validator** (não deve conter nenhum erro ou *warning*);
- O trabalho deve ser entregue até a data/hora definida pelo professor. Não deixe para enviar o trabalho nos últimos instantes, pois eventuais problemas relacionados à eventos adversos como instabilidade de conexão, congestionamento de rede etc., não serão aceitos como motivos para entrega da atividade por outras formas ou em outras datas;
- Este trabalho deve ser feito **mantendo os trabalhos anteriores intactos**, ou seja, os trabalhos anteriores devem permanecer online conforme foram entregues;
- Trabalhos enviados por e-mail ou pelo MS Teams **não serão considerados**.

Leia os slides de aula disponibilizados no endereço a seguir e resolva os exercícios seguintes.

<https://furtado.prof.ufu.br/site/teaching/PPI-GGI/PPI-Modulo8-Ajax-GGI.pdf>

### Exercício 1

---

Acesse o arquivo <https://furtado.prof.ufu.br/site/teaching/PPI/Exemplos-Ajax-XHR.zip> e coloque o exemplo **Ex3-recebendo-json** online. Teste o exemplo no navegador. Em seguida, acrescente comentários nos códigos JavaScript e PHP explicando as operações. Analise também os códigos do exemplo **Ex5-alunos-json**.

### Exercício 2

---

Crie um script PHP de nome **buscaEspecialidades.php** que retorne, no formato JSON, um array contendo os nomes de todas as especialidades médicas distintas dos médicos cadastrados na tabela **Medico** do banco de dados. Para resgatar facilmente todas as especialidades e armazená-

las em um array, utilize o método `fetchAll` conforme apresentado no material de aula. Lembre-se de converter o array para representação em JSON. Teste o script digitando o endereço completo (online) no navegador e observe o resultado.

### Exercício 3

---

Crie uma página HTML que apresente um campo do tipo **select** (inicialmente sem opções) para que o usuário possa selecionar uma especialidade médica. Ao invés de colocar as opções no **select** de forma estática, usando o próprio HTML, crie uma função JavaScript para buscar no servidor os nomes das especialidades disponíveis (de acordo com o cadastro médico) e inseri-las dinamicamente como opções do **select**. A função deve ser executada assim que a página for carregada. O código JavaScript deve fazer uma requisição Ajax ao script criado no exercício anterior. Para inserir as opções no **select** dinamicamente (retornadas pelo servidor como um array em JSON) será necessário manipular a árvore DOM de maneira similar àquela realizada no **Trabalho8/Exercício1** (usando funções e propriedades como **createElement**, **appendChild** e **textContent**).

### Exercício 4

---

Acrescente outro campo do tipo **select** no formulário para que o usuário possa selecionar um nome de médico. As opções do **select** devem ser carregadas automaticamente com JavaScript assim que o usuário escolher/trocar a especialidade médica no **select** anterior. Será necessário:

- criar um script PHP que receba como parâmetro pela URL o nome da especialidade médica e retorne, no formato JSON, um array de objetos contendo os nomes e códigos de todos os médicos daquela especialidade;
- criar uma função JavaScript para buscar no servidor os nomes e códigos dos médicos da especialidade escolhida pelo usuário e inseri-los dinamicamente como opções do segundo **select** (para que o usuário possa escolher um médico daquela especialidade). O **código** do médico deve ser inserido no atributo **value** do **option**, enquanto o **nome** deve ser inserido como **conteúdo** do **option**;

### Disponibilização Online

---

As páginas dos exercícios devem ser disponibilizadas online utilizando o subdomínio gratuito registrado anteriormente, porém em pasta própria (isto é, `seusubdominio.com/trabalhoX/ex1`, `seusubdominio.com/trabalhoX/ex2`, etc.). Não altere ou exclua as pastas dos trabalhos anteriores.

Acrescente um arquivo de nome **index.html** na pasta raiz do trabalho contendo links para as páginas dos exercícios.

### Entrega

---

Além da disponibilização online, a pasta raiz contendo as subpastas dos exercícios deve ser compactada no formato zip e enviada pelo Sistema Acadêmico de Aplicação de Testes (SAAT) até a data limite indicada pelo professor em sala de aula.

Adicione também um arquivo de nome **link.txt**, na pasta raiz, contendo a URL do trabalho online (para a pasta raiz do trabalho).

## Sobre Eventuais Plágios

---

Este é um trabalho individual. Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.